Software Paradigms SS 2015 4

Nachname	Vorname	Matrikelnummer
LORENZ	Peter	
ZIKO	Haris	

Exercise 1

1.

t = p(Y,f(X),a), t' = p(X,f(b), X)

θ = {X/Y}
t = p(Y,f(Y), a); t' = p(Y,f(b),Y)

$\theta = \theta \cup$ {Y/b}
t = p(b,f(b),a); t' = p(b,f(b), b)

Not unifiable, see rule 2a from Slides 7 - S17. There are different values, and no Variables.

2.

t = q(f(a),g(f(a)),Y), t' = q(X,g(X),f(g(a)))

Θ= {X/f(a)}
t = q(f(a),g(f(a)),Y), t' = q(f(a),g(f(a)),f(g(a)))

$\Theta = \Theta \cup$ {Y/f(g(a))}
t = q(f(a),g(f(a)),f(g(a))), t' = q(f(a),g(f(a)),f(g(a)))
□
MGU = {X/f(a),Y/f(g(a))}

3.

t = r(b,f(g(X)), Z) und t' = r(X,Z,f(g(a)))

Θ= {X/b}
r(b,f(g(b)), Z); r(b,Z,f(g(a)))

Θ= {Z/f(g(b))}
r(b,f(g(b)), f(g(b))); r(b,f(g(b)),f(g(a)))

MGU = {X/b, Z/f(g(b))}

Not unifiable. Slides 7 - S17. It exists a difference correspondig to 2.a, but there are not any variables. The algorithm stops, because there is a difference between t and t'.

4.

t = s(X,c,Y), t' = s(g(a,b),Z,f(Y))

Θ= {X/g(a,b)}
t = s(g(a,b),c,Y), t' = s(g(a,b),Z,f(Y))

Not unifiable, see rule 2b from Slides 7 - S17. It is not possible to unify Y with f(Y).

5.

t = u(f(Z,g(X)),Z), t' = u(f(g(a),g(Z)), X)

Θ= {Z/g(a)}
t = u(f(g(a),g(X)),g(a)), t' = u(f(g(a),g(g(a))), X)

Θ= Θ∪ {X/g(a)}
t = u(f(g(a),g(g(a))),g(a)), t' = u(f(g(a),g(g(a))), g(a))
□

MGU = {Z/g(a),X/g(a)}

Exercise 2

Predicate for greater(X,Y)
greater(s(_), 0) :=.
greater(s(X),s(Y)) := greater(X,Y)
show, if the next statement is true or false:
:=greater(s(s(0))), s(s(0))))
¬ greater(s(s(0))), s(s(0)))) ¬ greater(X,Y) ∨ greater(s(X),s(Y))
{X| s(0)),Y| s(0))}

¬ greater(s(s(0)), s(s(0))) ¬ greater(X,Y) ∨ greater(s(X),s(Y))
{X| s(0),Y| s(0)}

¬ greater(s(0), s(0)) ¬ greater(X,Y) ∨ greater(s(X),s(Y))
{X| 0,Y| 0}

¬ greater(0, 0) ¬ greater(X,Y) ∨ greater(s(X),s(Y))
Not possible, because no MGU exists.

¬ greater(0, 0) ¬ greater(s(_),0)
Not possible, because left side is not equal to right side.
Reject input =¿ False

Exercise 3

1.

```
male(mannister).

male(lombard).

male(tytos).
female(genna).

male(jamballalia).

male(kevan).
male(lancel).
male(willem).
```

```
female(janei).

male(tywin).
male(jaime).
female(cersei).
male(tyrion).

parent(mannister, lombard).
parent(mannister, tytos).
parent(mannister, jamballalia).

parent(tytos, genna).
parent(tytos, tywin).
parent(tytos, kevan).

parent(kevan, lancel).
parent(kevan, willem).
parent(kevan, janei).

parent(tywin, jaime).
parent(tywin, cersei).
parent(tywin, tyrion).
```

2.

```
grandp(X,Y)  :- parent(X, P), parent(P, Y).
sibling(X,Y) :- parent(P,X), parent(P,Y), X\=Y.
cousin(X,Y)  :- parent(A,X), parent(B,Y), sibling(A,B), male(X).
```

c1 = grandp(X,Y) $\vee$ $\neg$parent(X, P) $\vee$ $\neg$parent(P, Y).
c2 = sibling(X,Y) $\vee$ $\neg$parent(P,X) $\vee$ $\neg$parent(P,Y) $\vee$ $\neg X\bar{Y}$.
c3 = cousin(X,Y) $\vee$ $\neg$ male(X) $\vee$ $\neg$parent(A,X) $\vee$ $\neg$parent(B,Y) $\vee$ $\neg$ sibling(A,B).

:= cousin(willem, jaime).

Program	MGU	Rule
¬cousin(willem, jaime)	θ = {X/willem, Y/jaime}	cousin(X,Y) ∨ ¬ male(X) ∨ ¬parent(A,X) ∨ ¬parent(B,Y) ∨ ¬ sibling(A,B)
¬ male(willem) ∨ ¬parent(A,willem) ∨ ¬parent(B,jaime) ∨ ¬ sibling(A,B)		male(willem)
¬parent(A,willem) ∨ ¬parent(B,jaime) ∨ ¬ sibling(A,B)	θ = {A/kevan, B/tywin}	parent(kevan, willem) and parent(tywin, jaime)
sibling(kevan,tywing)	θ = {X/kevan, B/tywin}	sibling(X,Y) ∨ ¬parent(P,X) ∨ ¬parent(P,Y) ∨ ¬X$\bar{Y}$.
¬parent(P,X) ∨ ¬parent(P,Y) ∨ ¬X$\bar{Y}$	θ = {P/tytos}	parent(tytos, kevan) and parent(tytos, tywan)
□		

3.

granddaughter(X,Y) :- grandp(Y,X), female(X).

Program	MGU	Rule
:= granddaughter(X,tytos)	θ = {Y/tytos}	granddaughter(X,Y) ∨ ¬grandp(Y,X) ∨ female(X)
¬grandp(tytos,X) ∨ ¬female(X)	θ = {X/tytos}	grandp(X,Y) ∨ ¬parent(X,P) ∨ ¬parent(P,Y)
¬parent(tytos, P) ∨ ¬p(P,Y)	θ = {P/kevan}	parents(tytos, kevan).
¬parents(tytos, kevan) ∨ ¬parents(kevan,Y)	θ = {Y/janei}	p(kevan, janei)
parents(kevan, janei) □		

Exercise 4

1.

edge(a, b, s(s(s(0)))).
edge(a, c, s(0)).
edge(c, d, s(s(0))).
edge(b, d, s(s(0))).

add(0,Y,Y). add(s(X),Y,s(Z)) := add(X,Y,Z). greaterequ(X,0).
greaterequ(s(X), s(Y)) := greaterequ(X,Y).

path(S,E,M) := find(S,E,N),greaterequ(M,N).

find(S,E,L) := edge(S,E,L).
find(S,E,N) := edge(S,X,L),find(X,E,N1),add(N1,L,N).

2.

c1: edge(a, b, s(s(s(0)))).
c2: edge(a, c, s(0)).
c3: edge(c, d, s(s(0))).
c4: edge(b, d, s(s(0))).
c5: add(0,Y,Y). c6: add(s(X),Y,s(Z)) ∨ ¬ add(X,Y,Z). c7: greaterequ(X,0).
c8: greaterequ(s(X), s(Y)) ∨ ¬ greaterequ(X,Y).
c9: path(S,E,M) ∨ ¬ find(S,E,N) ∨ ¬ greaterequ(M,N).
c10: find(S,E,L) ∨ ¬ edge(S,E,L).
c11: find(S,E,N) ∨ ¬ edge(S,X,L) ∨ ¬ find(X,E,N1) ∨ ¬ add(N1,L,N)

Program	MGU	Rule
¬ path(a,d,s(s(s(s(0)))))	{S/a, E/d, M/s(s(s(s(0))))}	c9: path(S,E,M) ∨ ¬ find(S,E,N) ∨ ¬ greaterequ(M,N)
¬ find(a,d,N) ∨ ¬ greaterequ(s(s(s(s(0)))),N)	{S/a,E/d}	c11: find(S,E,N) ∨ ¬ edge(S,X,L) ∨ ¬ find(X,E,N1) ∨ ¬ add(N1,L,N)
¬ edge(a,X,L) ∨ ¬ find(X,d,N1) ∨ ¬ add(N1,L,N) ¬ greaterequ(s(s(s(s(0)))),N)	{X/c,L/s(0)}	c2: edge(a, c, s(0))
¬ find(c,d,N1) ∨ ¬ add(N1,s(0),N) ¬ greaterequ(s(s(s(s(0)))),N)	{S/c,E/d/L/N1}	c10: find(S,E,L) ∨ ¬ edge(S,E,L)
¬ edge(c,d,N1) ∨ ¬ add(N1,s(0),N) ¬ greaterequ(s(s(s(s(0)))),N)	{N1/s(s(0))}	c3: edge(c, d, s(s(0)))
¬ add(s(s(0)),s(0),N) ∨ ¬ greaterequ(s(s(s(s(0)))),N)	{X/s(0),Y/s(0),N/s(Z)}	c6: add(s(X),Y,s(Z)) ∨ ¬ add(X,Y,Z)
¬ add(s(0),s(0),s(Z)) ∨ ¬ greaterequ(s(s(s(s(0)))),s(Z))	{X'/s(0),Y'/s(0),Z/s(Z')}	c6: add(s(X'),Y',s(Z')) ∨ ¬ add(X',Y',Z')
¬ add(0,s(0),s(s(Z'))) ∨ ¬ greaterequ(s(s(s(s(0)))),s(s(Z')))	{Y"/s(0),Z'/Y"}	c5: add(0,Y",Y")
¬ greaterequ(s(s(s(s(0)))),s(s(s(0))))	{X/s(s(s(0))),Y/s(s(s(0)))}	c8: greaterequ(s(X), s(Y)) ∨ ¬ greaterequ(X,Y)
¬ greaterequ(s(s(s(0))),s(s(0)))	{X'/s(s(0)),Y'/s(s(0))}	c8: greaterequ(s(X'), s(Y')) ∨ ¬ greaterequ(X',Y')
¬ greaterequ(s(s(0)),s(0))	{X"/s(0),Y"/s(0)}	c8: greaterequ(s(X"), s(Y")) ∨ ¬ greaterequ(X",Y")
¬ greaterequ(s(0),0)	{X"'/s(0)}	c7: greaterequ(X"',0)
□		

We can say, that the input

Exercise 5

1.

less(0, s(Y)).
less(s(X),s(Y)) := less(X,Y).

union(null,null,null).
union(null,Y,Y).
union(X,null,X).

union(build(X,Xs),null,build(X,Zs)) := union(Xs,null,Zs).
union(null,build(Y,Ys),build(Y,Zs)) := union(null,Ys,Zs).

union(build(Z,Xs),build(Z,Ys),build(Z,Zs)) := union(Xs,Ys,Zs).
union(build(X,Xs),build(Y,Ys),build(X,Zs)) := less(X,Y), union(Xs,build(Y,Ys),Zs).
union(build(X,Xs),build(Y,Ys),build(Y,Zs)) := less(Y,X), union(build(X,Xs),Ys,Zs).

2.

	Resolution for Example 5 Nr. 2
Program	:= union(build(0, build(s(s(0)), null)), build(s(0) , null), build(0, build(s(0), build(s(s(0)), null)))).
Rule	union(build(X,XS),build(Y,Ys),build(X,Zs)) := less(X,Y), union(XS,build(Y,YS),Zs).
MGU	{X/0,Xs/build(s(s(0)), null),Y/s(0),Ys/null,Zs/build(s(0), build(s(s(0)), null))}
Program	:= less(0,s(0)), union(build(s(s(0)), null),build(s(0),null),build(s(0), build(s(s(0)), null))).
Rule	less(0, s(Y)).
MGU	{Y/0}
Program	:= union(build(s(s(0)), null),build(s(0),null),build(s(0), build(s(s(0)), null))).
Rule	union(build(X,Xs),build(Y,Ys),build(Y,Zs)) := less(Y,X), union(build(X,Xs),Ys,Zs).
MGU	{X/s(s(0)),Xs/null,Y/s(0),Ys/null,Zs/build(s(s(0)), null)}
Program	:= less(s(0),s(s(0))), union(build(s(s(0)),null),null,build(s(s(0)), null)).
Rule	less(s(X),s(Y)) := less(X,Y).
MGU	{X/0,Y/s(0)}
Program	:= less(0,s(0)), union(build(s(s(0)),null),null,build(s(s(0)), null)).
Rule	less(0, s(Y)).
MGU	{Y/0}
Program	:= union(build(s(s(0)),null),null,build(s(s(0)), null)).
Rule	union(build(X,Xs),null,build(X,Zs)) := union(Xs,null,Zs).
MGU	{X/s(s(0)),Xs/null,Zs/null}
Program	:= union(null,null,null).
Rule	union(null,null,null).
MGU	{}
Program	□